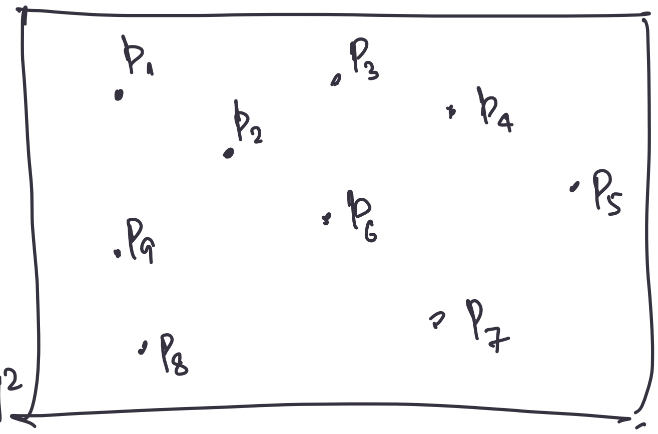
Closest pair of points

Input: $p_1, \dots, p_n$ ~~and distances between them.~~

(x_1, y_1) (x_2, y_2)

↳ $\sqrt{|x_2-x_1|^2+|y_2-y_1|^2}$



→ Distance metric defined.

$d(p_i, p_j) > 0$ if $i \neq j$

$d(p_i, p_i) = 0$.

Problem from Computational Geometry

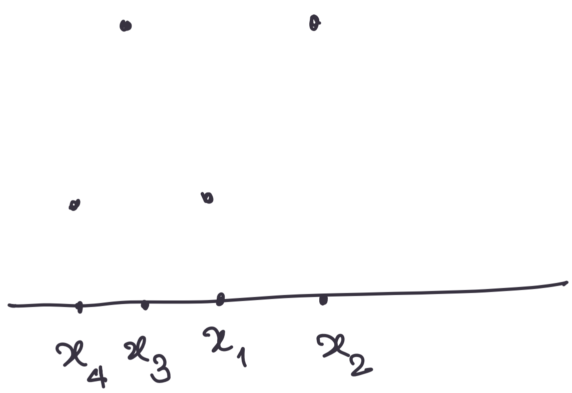
Want: Find the pair that has least distance between them.

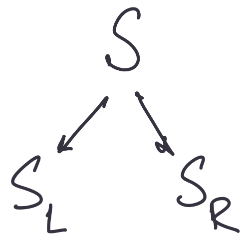
Algo 1: Look at $d(p_i, p_j)$ for all pairs $i \neq j$.
$O(n^2)$ time.

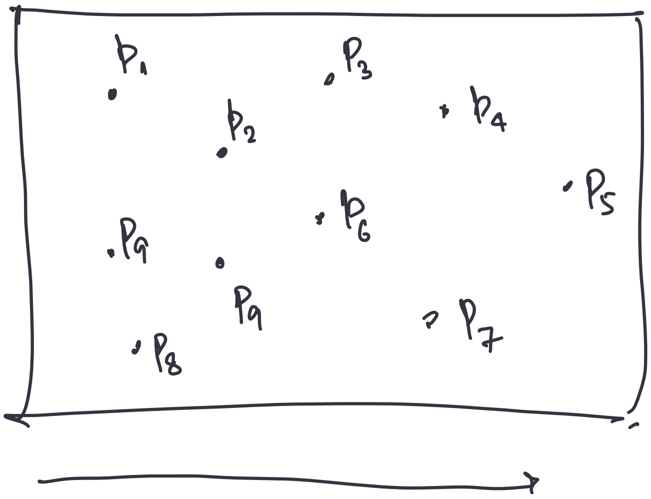
1-dimensional version: Find closest pair of points on a line.

↳ Sort the points and then find closest successive points
$\underbrace{\text{Sort the points}}_{O(n\log n)}$ $\underbrace{\text{successive}}_{O(n)}$.

→ Sort first wrt x-coordinates and then y-coordinates if there is a tie wrt x-coordinates.







Sort the points wrt x- coordinates.

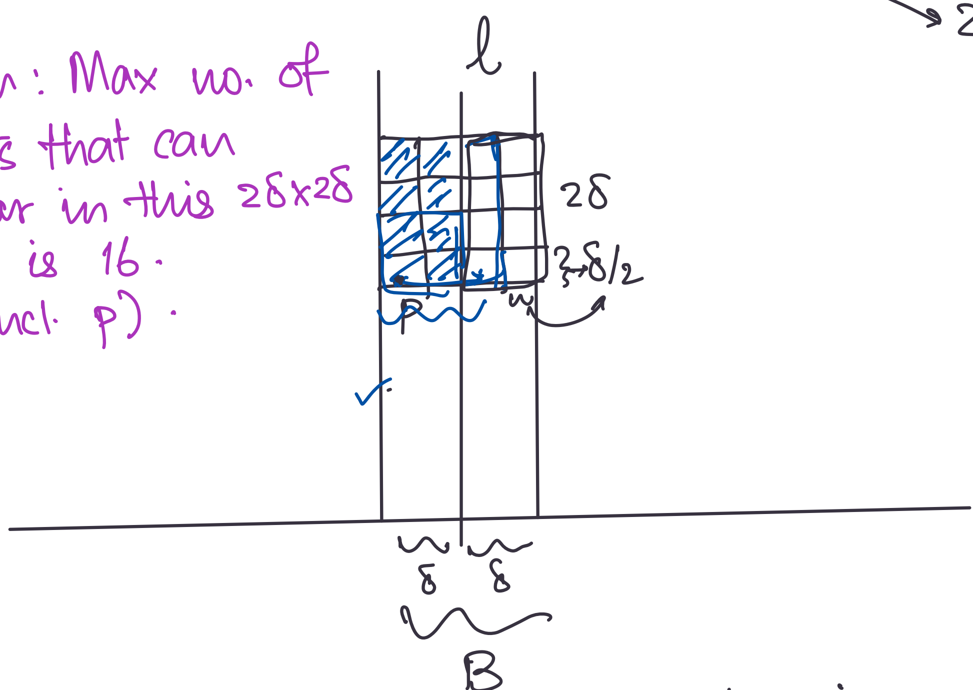
Get a solution to closest pair of points in S_L : p_1^L, p_2^L
and " in S_R : p_1^R, p_2^R

$\delta = \min\{\underline{\underline{d(p_1^L, p_2^L)}}, d(p_1^R, p_2^R)\}.$

attains the min.

1. p_1^L, p_2^L is infact a solution.
2. $\exists$ points (p, q) s.t. $p \in S_L$, $q \in S_R$ and $d(p,q) < \underline{\underline{\delta}}$.
   They will appear in the band B.

Claim: Max no. of points that can appear in this $2\delta \times 2\delta$ grid is 16. (incl. p).



• Sort the points in B in the incr. order of y coordinates

Claim: Given a point $p \in B$, there are at most 15 points $q \in B$ s.t. y coordinate of $q \geq$ y coordinate of p and $d(p,q) < \delta$.

→ Max distance between any pair of points in a square of size $\frac{\delta}{2} \times \frac{\delta}{2}$ is $\frac{\delta}{2} \times \sqrt{2} < \delta$

If two points exist in the same square, their dist is $< \delta$ and this violates the fact that δ was the min dist in L & R.

→ Any point above the grid has dist strictly larger than δ.

For each p in the y-coord sorted list of points in B:

compute distances of p to its 15 successive points.

Take min over all these distances.

If the min is $< \delta$ then report that corresponding pair as the closest pair of points

Else, report P_1^L and P_2^L as closest pair.

Algo:

→ Sort the given points in incr order of x-coordinates.

→ Break the problem into two parts by drawing a line l ($\perp$ x axis) in the middle.

→ Obtain solution recursively in both the "halves".

Let δ be the min of both the solutions.

→ B ← set of points that are at most δ dist away from

ℓ (x-coordinate wise)

→ Sort the points in B in incr. order of y co-ordinate.

→ For each p in the sorted order:

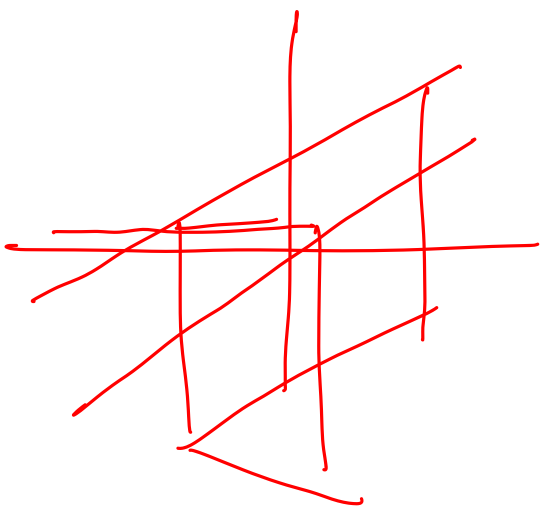
compute dist of p from 15 succ. points

→ Take min of all these distances.

If this min $< \delta$ then report this corr. pair as the closest pair of points

Else, report the min solution obtained from recursion.

Running time → $O(n\log n)$.



Iterated Matrix Product
Integer

$M_1, M_2, \ldots, M_d$ each $\underline{n \times n}$

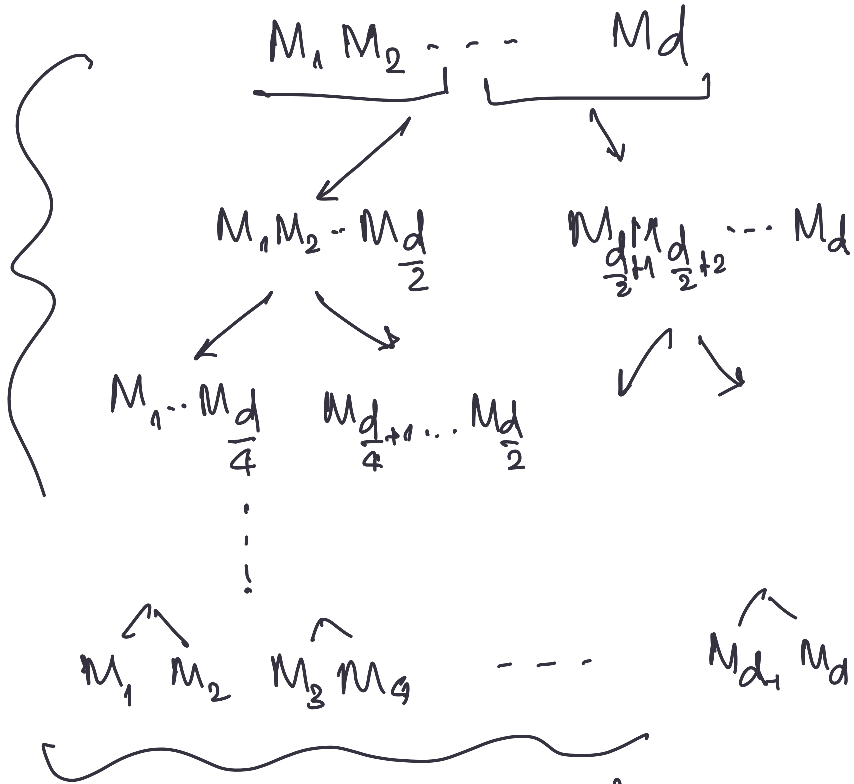
Simple algo:

$Acc \leftarrow I_{n \times n}$

For i in $[1, d]$:

$\quad Acc \leftarrow Acc \times M_i$

return Acc.



no. of mult is d.

---

no. of mult $\leftarrow d$

<u>Seq</u>: We have one processor computing d matrix mult.

Parallel: We can take $\frac{d}{2}$ processors and each processor needs to do at most $\log_2 d$ matrix mult.